

C++ 프로그래밍 및 실습

University couple

진척 보고서 #3

제출일자 : 2023.12.10

제출자명 : 서현호

제출자학번 : 203566

1. 대학생들의 삶의 질 향상 및 추억 제공

1) 배경 및 필요성

코로나로 인해 대학생활을 하면서 즐길 수 있는 여러가지 활동들을 못하게 되었고 취업난으로 인해 자격증과 공부에 몰입할 수밖에 없는 구조에서 학생들은 인생의 한 번 뿐인 대학생활을 즐기지 못하고 삭막한 사회로 들어갈 위기에 처해져 있음. 특히 공대의 경우 남학생들의 비율이 압도적으로 많고 또 교육과 같은 경우에도 여학생들의 비율이 많음. 이러한 학생들은 연애를 시도하기조차 어렵고 비교적 이성을 만나기 쉬운 학생들도 연애에 더욱 신중해질 수밖에 없다. 그렇게 되면서 점점 캠퍼스의 낭만을 즐길 여유가 사라지고 있다. 그리하여 학생들에게 연애 할 기회를 주기 위해서 프로그램을 계획하게 되었음.

2) 프로젝트 목표

학생들의 성격과 연애관을 바탕으로 모집된 표본들 사이에서 일정 수준 이상의 연애관 일치가 존재하거나, 학생들이 좋아하는 성격을 서로 갖고 있다면 매칭을 해주어 이제 곧 사회 진출하기 직전의 전남대학교 학생들에게 캠퍼스의 낭만을 즐기게 해줄 수 있는 기회를 주는 것이 목표.

3) 차별점

기존의 매칭 프로그램들은 단순히 사진을 게시하고 간단한 채팅으로 서로를 알아 보는, 가벼운 만남을 할 수 밖에 없는 구조를 가진 프로그램들만 존재한다. 또한 불특정 다수의 알 수 없는 사람들이 모두 참여하는 프로그램과 달리 전남대학생 들을 대상으로 하였고 성격과 연애관을 중심으로 보아 기존의 가벼운 만남에서 탈피한 진중한 연애로 발전할 가능성이 더 높은 매칭 프로그램이다

2. 기능 계획

1) 기능 1 - 프로그램에 참여한 학생들의 정보를 입력 받기

(1) 세부 기능 1

- 학생들의 개인정보(인스타 아이디)를 입력 받는다

(1) 세부 기능 2

- 30개의 연애상황을 주고 그 상황에서 본인의 행동이 어떤 지에 대한 보기를 4개 주고 그에 따른 학생들의 답변을 입력 받는다.

2) 기능 2

- 학생들에게 입력 받은 데이터(답변)를 통해 5가지 항목에 대한 점수를 매긴다. (학생들의 답변마다 5가지 항목에 대한 점수 기여도가 다르다.)

3) 기능 3

- 5가지 항목을 통해 분류된 학생들 간의 연애관 및 성격의 유사성을 비교하여 퍼센트로 환산한다.

4) 기능 4

- 학생들 간의 연애관에 따른 상호보완성을 고려한 가산점을 부여한다.

5) 기능 5

- 환산한 퍼센트와 가산점을 합산하여 일정 수준 이상의 매칭 점수를 받은 학생들을 뽑아서 매칭을 시켜준다

3. 진척사항

1) 기능 구현

(1) 기능1(프로그램에 참여한 학생들의 정보를 입력받기)

- 입력 : 학생들의 이름, 인스타 아이디, 성별을 입력받음
- 설명 : 프로그램을 시작하기에 앞서 학생들의 개인정보를 입력받는 기능
- 적용된 배운 내용

1. Student라는 클래스를 형성하고 클래스로 student 객체를 만들.

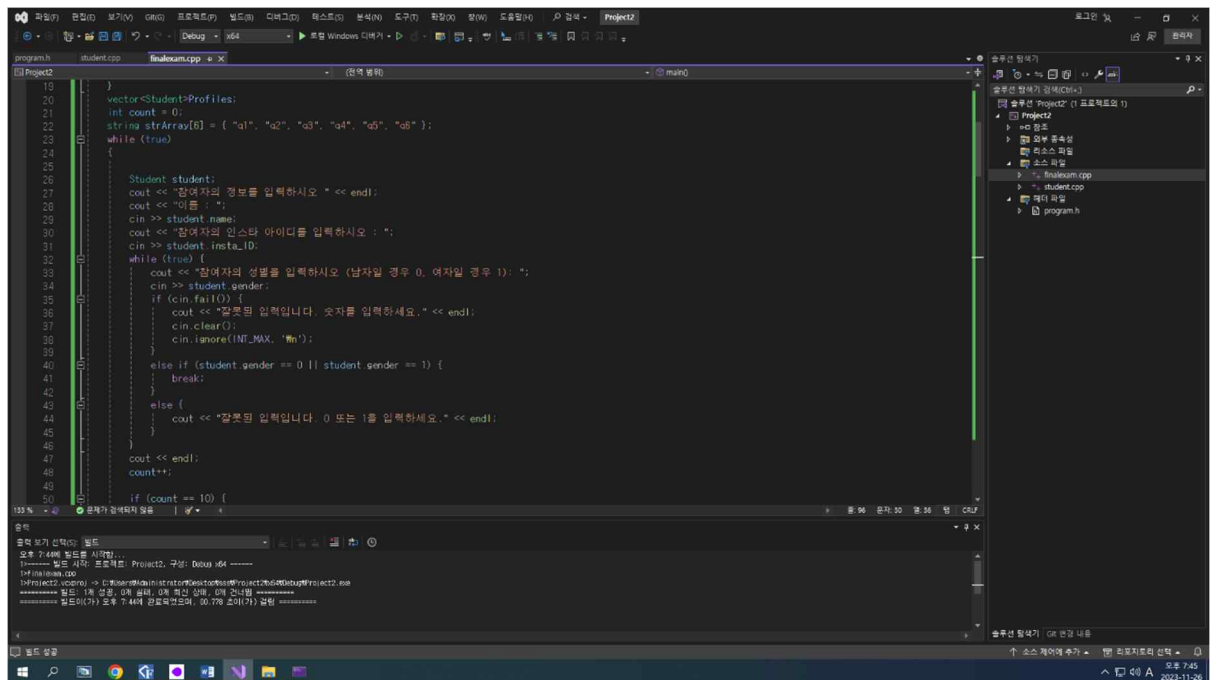
2. 만든 student 객체들을 vector Profiles에 하나씩 저장하여 학생들의 개인정보들 및 프로그램에 필요한 데이터들을 객체로 묶어서 관리하고 참여하는 학생들의 수를 자유롭게 함.

3. 반복문 while문을 사용하여 학생들의 성별을 판단할 때 잘못된 정보를 입력했을 시 다시 입력하도록 하는 경고문을 작성함.

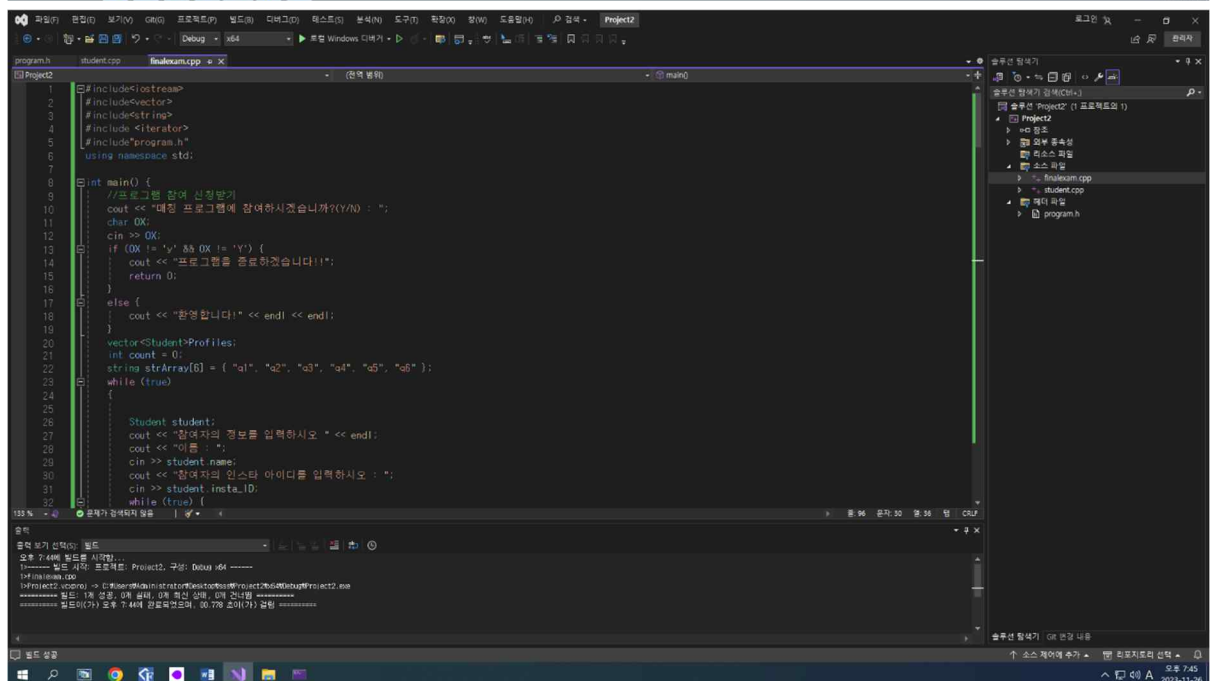
4. 조건문 if문을 사용하여 프로그램의 필요한 최소한의 학생 수가 모두 채워졌을 시 더 추가 할 것 인지를 물어봄.

5. push_back 함수를 사용하여 student 객체들을 Profiles 벡터에 넣는다.

-코드 스크린샷



```
19 vector<Student>Profiles;
20 int count = 0;
21 string strArray[6] = { "a1", "a2", "a3", "a4", "a5", "a6" };
22 while (true)
23 {
24     Student student;
25     cout << "참여자의 정보를 입력하십시오 : " << endl;
26     cout << "이름 : ";
27     cin >> student.name;
28     cout << "참여자의 인스타 아이디를 입력하십시오 : ";
29     cin >> student insta_ID;
30     while (true) {
31         cout << "참여자의 성별을 입력하십시오 (남자일 경우 0, 여자일 경우 1) : ";
32         cin >> student gender;
33         if (cin.fail()) {
34             cout << "잘못된 입력입니다. 숫자를 입력하세요." << endl;
35             cin.clear();
36             cin.ignore(INT_MAX, '\n');
37         }
38         else if (student gender == 0 || student gender == 1) {
39             break;
40         }
41         else {
42             cout << "잘못된 입력입니다. 0 또는 1을 입력하세요." << endl;
43         }
44     }
45     cout << endl;
46     count++;
47     if (count == 10) {
48         if (count == 10) {
49             if (count == 10) {
```

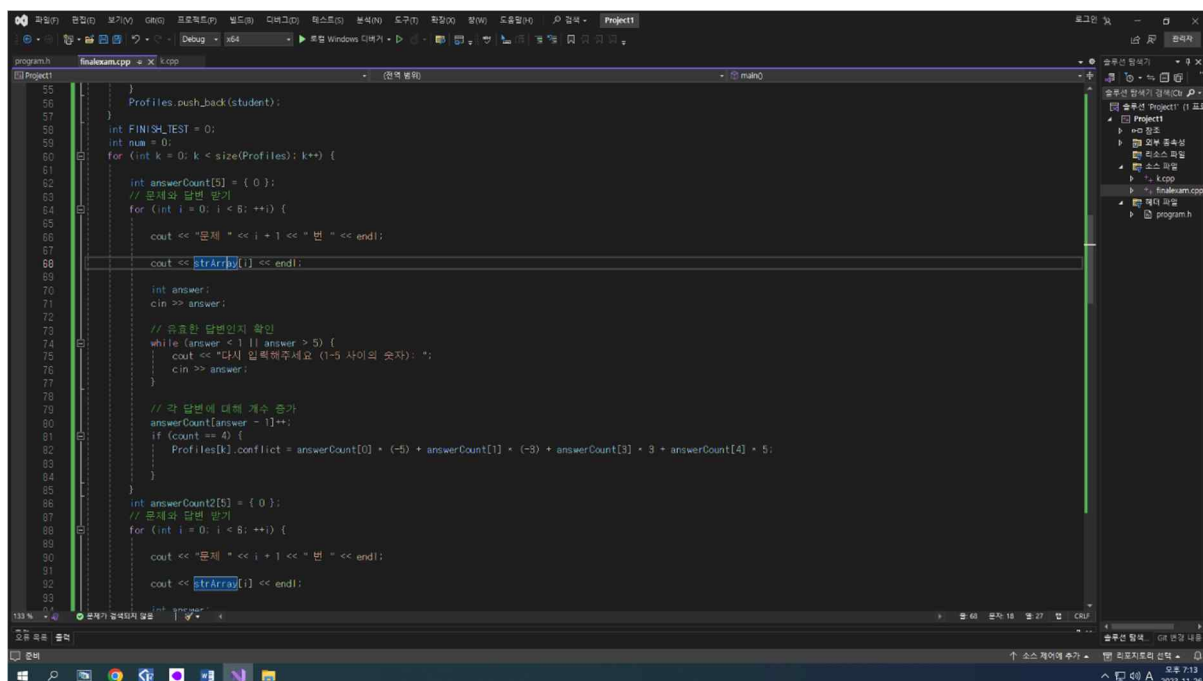


```
1 #include <iostream>
2 #include <vector>
3 #include <string>
4 #include <iterator>
5 #include "program.h"
6 using namespace std;
7
8 int main() {
9     //프로그램 참여 신청받기
10    cout << "대정 프로그램에 참여하시겠습니까?(Y/N) : ";
11    char OK;
12    cin >> OK;
13    if (OK != 'y' && OK != 'Y') {
14        cout << "프로그램을 종료하겠습니다!!";
15        return 0;
16    }
17    else {
18        cout << "환영합니다!" << endl << endl;
19    }
20    vector<Student>Profiles;
21    int count = 0;
22    string strArray[6] = { "a1", "a2", "a3", "a4", "a5", "a6" };
23    while (true)
24    {
25        Student student;
26        cout << "참여자의 정보를 입력하십시오 : " << endl;
27        cout << "이름 : ";
28        cin >> student.name;
29        cout << "참여자의 인스타 아이디를 입력하십시오 : ";
30        cin >> student insta_ID;
31        while (true) {
32            cout << "참여자의 성별을 입력하십시오 (남자일 경우 0, 여자일 경우 1) : ";
33            cin >> student gender;
34            if (cin.fail()) {
35                cout << "잘못된 입력입니다. 숫자를 입력하세요." << endl;
36                cin.clear();
37                cin.ignore(INT_MAX, '\n');
38            }
39            else if (student gender == 0 || student gender == 1) {
40                break;
41            }
42            else {
43                cout << "잘못된 입력입니다. 0 또는 1을 입력하세요." << endl;
44            }
45        }
46        cout << endl;
47        count++;
48        if (count == 10) {
49            if (count == 10) {
50                if (count == 10) {
```

(2) 기능2(학생들에게 입력 받은 데이터를 통해 5가지 항목에 대한 점수를 매기고 계산한다.)

- 입력 : 학생들의 문제에 대한 답변을 입력받음
- 설명 : 각 문제에 대한 학생들의 답변을 입력받고 답변에 따른 점수를 매긴 후 학생들의 클래스에 저장하는 기능
- 적용된 배운 내용
 1. 배열 answerCount를 선언 후 0으로 초기화 한 후 학생들의 답변을 answerCount라는 배열에 입력받아 사용함
 2. 반복문 for문을 사용하여 배열 answerCount에 저장되어 있는 문제들을 하나씩 출력한다.
 3. 반복문 while문을 사용하여 유효한 답변인지 확인 후 유효하지 않을 경우 다시 입력하도록 제안한다.

-코드 스크린 샷



```
55 }
56 Profiles.push_back(student);
57 }
58 int FINISH_TEST = 0;
59 int num = 0;
60 for (int k = 0; k < size(Profiles); k++) {
61
62     int answerCount[5] = { 0 };
63     // 문제와 답변 받기
64     for (int i = 0; i < 5; ++i) {
65
66         cout << "문제 " << i + 1 << " 번 " << endl;
67
68         cout << strArray[i] << endl;
69
70         int answer;
71         cin >> answer;
72
73         // 유효한 답변인지 확인
74         while (answer < 1 || answer > 5) {
75             cout << "다시 입력해주세요 (1-5 사이의 숫자): ";
76             cin >> answer;
77         }
78
79         // 각 답변에 대해 계수 증가
80         answerCount[answer - 1]++;
81         if (count == 4) {
82             Profiles[k].conflict = answerCount[0] * (-5) + answerCount[1] * (-3) + answerCount[3] * 3 + answerCount[4] * 5;
83         }
84     }
85
86     int answerCount2[5] = { 0 };
87     // 문제와 답변 받기
88     for (int i = 0; i < 5; ++i) {
89
90         cout << "문제 " << i + 1 << " 번 " << endl;
91
92         cout << strArray[i] << endl;
93     }
```

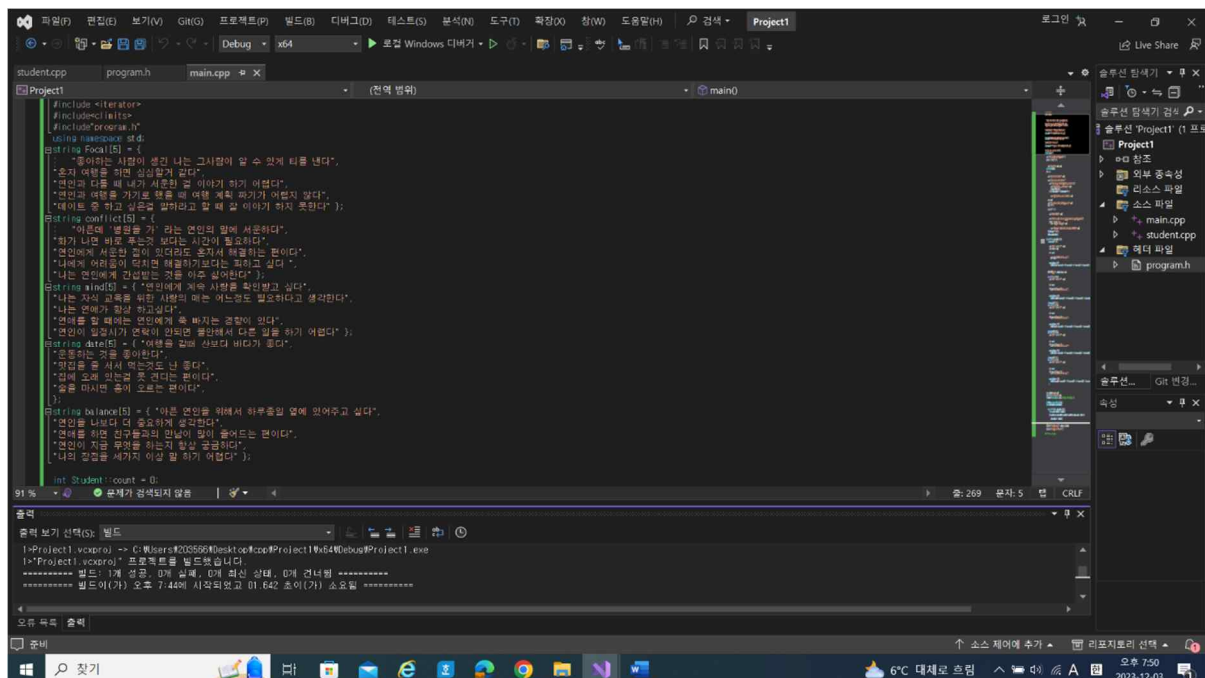
(3) 기능2(학생들에게 할 질문 선정)

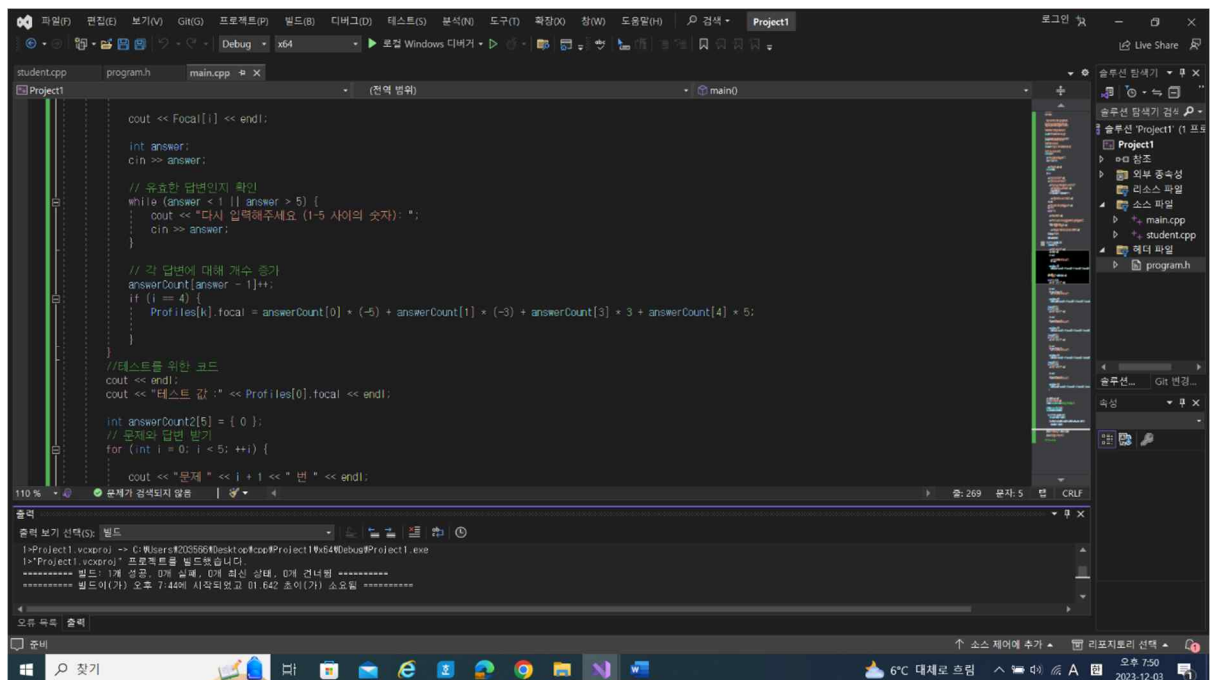
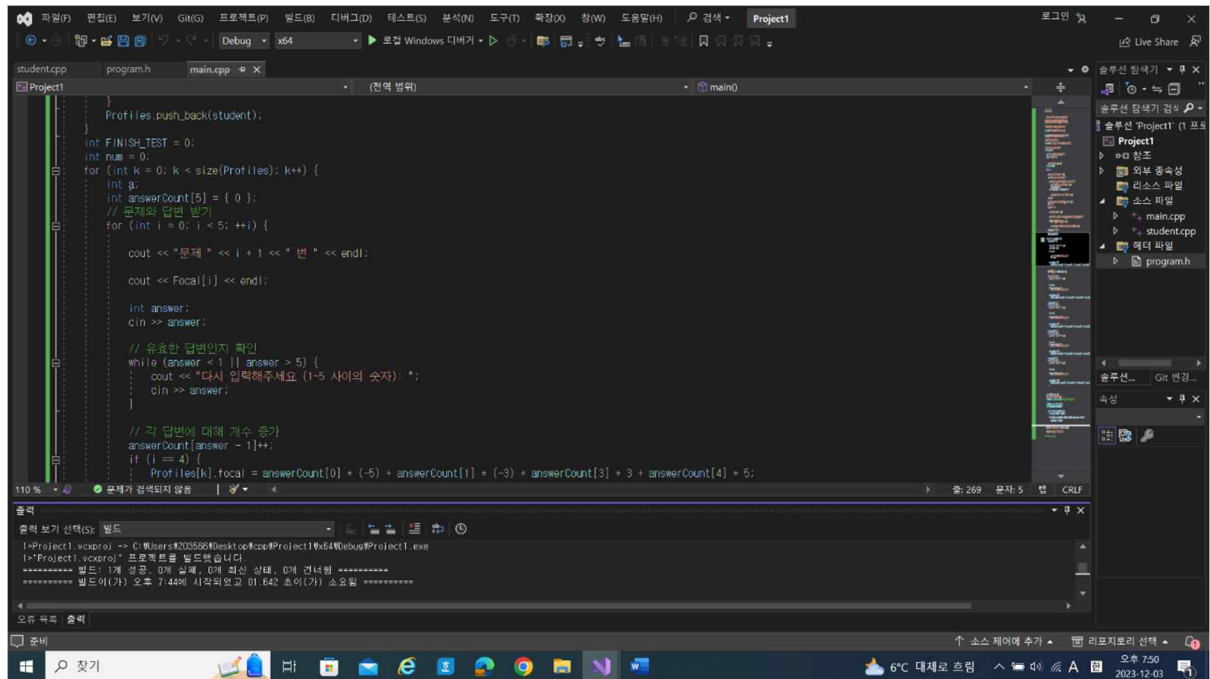
- 입출력 : 학생들에게 할 질문 입력 및 출력
- 설명 : 커플 선정에 필요한 데이터를 얻기 위해 할 질문들을 선정 후 학생들에게 질문한 후 답변을 저장한다
- 적용된 배운 내용

1. 5가지 항목에 대한 질문들을 배열에 저장하였고 배열을 반복문을 통해서 다시 출력하도록 하였다.

2. 받은 답변을 answerCount라는 배열에 집어넣고 다시 그 배열의 값들을 계산하여 Profiles 벡터에 집어 넣는다.

-코드 스크린샷





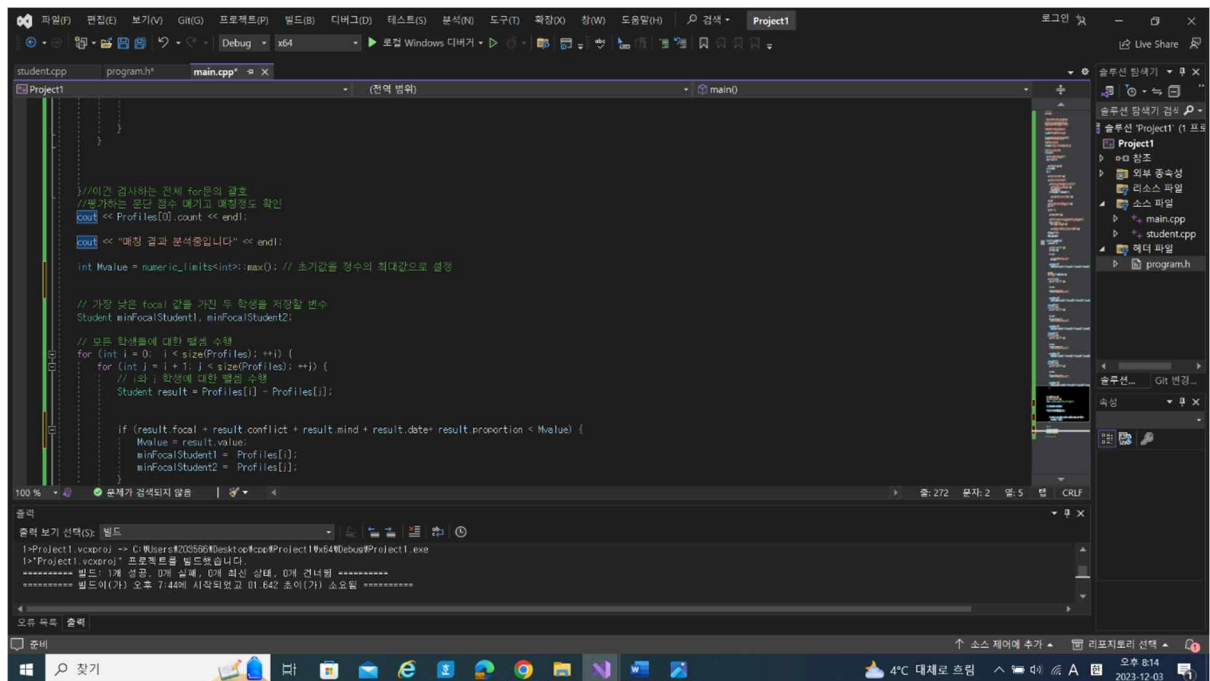
(4) 기능3 (학생들에게 받은 데이터를 통해 점수를 계산)

- 입력 : 학생들의 답변 결과로 얻은 점수, 점수들끼리의 결합
- 설명 : 학생들의 답변 결과로 생성된 5가지의 항목에 대한 점수들을 특정 방법으로 계산하여 좋은 커플이 될 수 있는 가능성이 가장 높은 두 명을 선정하여 뽑았다.

- 적용된 배운 내용

1. operator 연산자 함수를 통해 클래스 내의 멤버변수들의 차를 계산하고 계산하였다.
2. 반복문을 사용하여 배열들을 순차적으로 계산식에 집어넣어 모든 참가자들끼리 공합을 볼 수 있도록 코드를 구성하였다.

-코드 스크린 샷



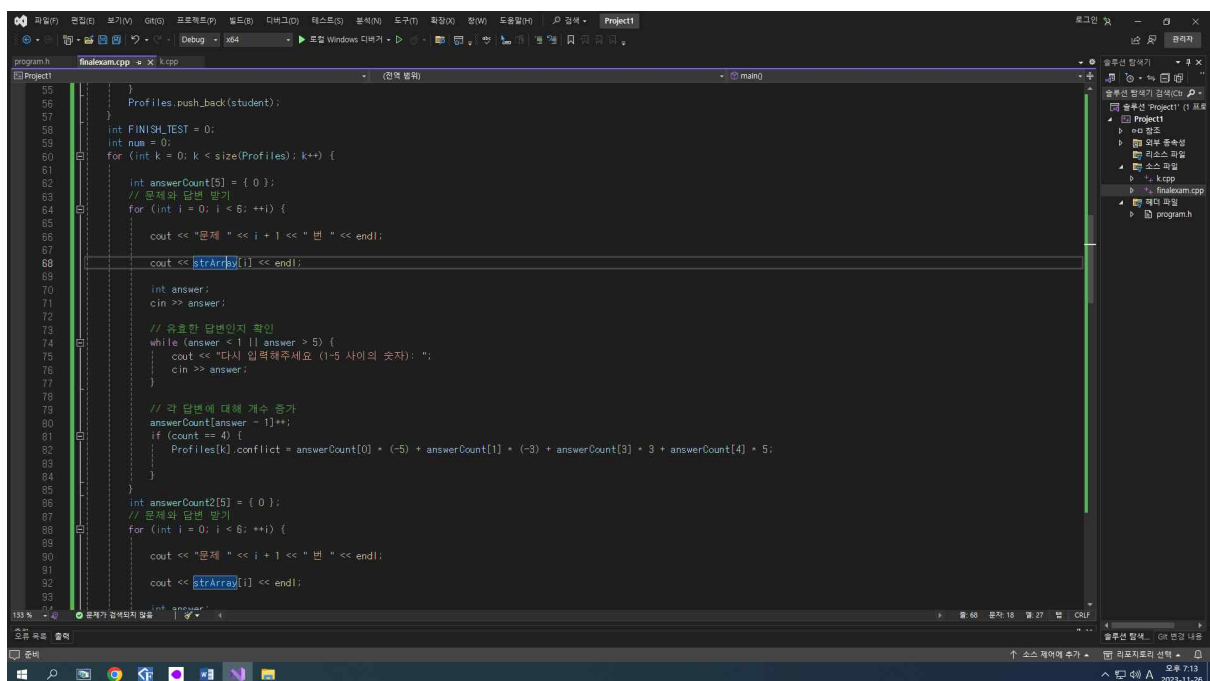
```
//이런 검사하는 전체 for문의 끝은
//꼭가하는 문단 점수 빼고 매정정도 확인
cout << Profiles[0].count << endl;
cout << "매정 결과 분석중입니다" << endl;

int Mvalue = numeric_limits<int>::max(); // 초기값을 정수의 최대값으로 설정

// 가장 낮은 focal 값을 가진 두 학생을 저장할 변수
Student minFocalStudent1, minFocalStudent2;

// 모든 학생들에 대한 탐색 수행
for (int i = 0; i < size(Profiles); ++i) {
    for (int j = i + 1; j < size(Profiles); ++j) {
        // (i, j) 학생에 대한 탐색 수행
        Student result = Profiles[i] - Profiles[j];

        if (result.focal + result.conflict + result.mind + result.data + result.proportion < Mvalue) {
            Mvalue = result.value;
            minFocalStudent1 = Profiles[i];
            minFocalStudent2 = Profiles[j];
        }
    }
}
```



```
Profiles.push_back(student);

int FINISH_TEST = 0;
int num = 0;
for (int k = 0; k < size(Profiles); k++) {

    int answerCount[5] = { 0 };
    // 문제와 답변 받기
    for (int i = 0; i < 5; ++i) {

        cout << "문제 " << i + 1 << " 번 " << endl;
        cout << strArray[i] << endl;

        int answer;
        cin >> answer;

        // 유효한 답변인지 확인
        while (answer < 1 || answer > 5) {
            cout << "다시 입력해주세요 (1-5 사이의 숫자): ";
            cin >> answer;
        }

        // 각 답변에 대해 계수 증가
        answerCount[answer - 1]++;
        if (count == 4) {
            Profiles[k].conflict = answerCount[0] * (-5) + answerCount[1] * (-3) + answerCount[3] * 3 + answerCount[4] * 5;
        }
    }

    int answerCount2[5] = { 0 };
    // 문제와 답변 받기
    for (int i = 0; i < 5; ++i) {

        cout << "문제 " << i + 1 << " 번 " << endl;
        cout << strArray[i] << endl;

        int answer;
        cin >> answer;

        // 유효한 답변인지 확인
        while (answer < 1 || answer > 5) {
            cout << "다시 입력해주세요 (1-5 사이의 숫자): ";
            cin >> answer;
        }

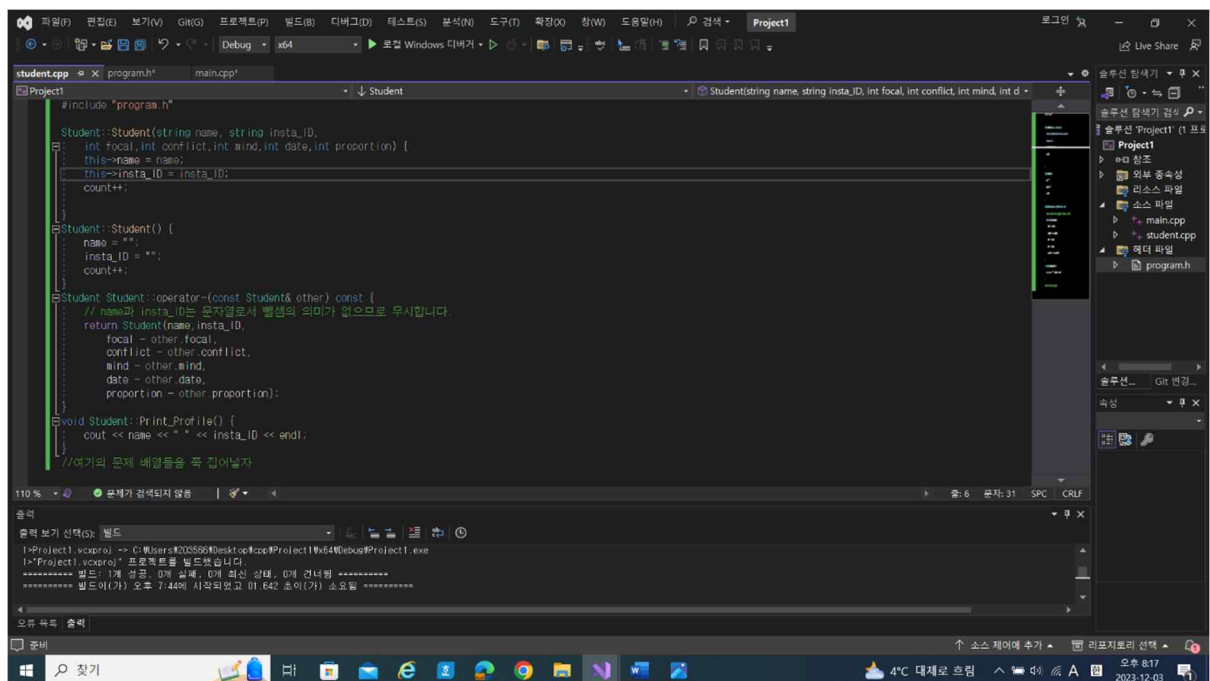
        // 각 답변에 대해 계수 증가
        answerCount2[answer - 1]++;
        if (count == 4) {
            Profiles[k].proportion = answerCount2[0] * (-5) + answerCount2[1] * (-3) + answerCount2[3] * 3 + answerCount2[4] * 5;
        }
    }
}
```

(5) 추가 기능(학생들의 총 수 구하기)

- 입출력 : 생성된 객체의 개수
- 설명 : 객체가 생성될 때 마다 늘어나게 하여 총 학생 수를 구한다.
- 적용된 배운 내용

1. count라는 static 변수를 클래스 내부에 생성하여 객체들이 생성될때 마다 늘어나는 변수를 만들었다.

-코드 스크린 샷-



```
#include "program.h"

Student::Student(string name, string insta_ID,
    int focal, int conflict, int mind, int date, int proportion) {
    this->name = name;
    this->insta_ID = insta_ID;
    count++;
}

Student::Student() {
    name = "";
    insta_ID = "";
    count++;
}

Student Student::operator=(const Student& other) const {
    // name과 insta_ID는 문자열로서 배열의 의미가 없으므로 무시합니다.
    return Student(name, insta_ID,
        focal = other.focal,
        conflict = other.conflict,
        mind = other.mind,
        date = other.date,
        proportion = other.proportion);
}

void Student::Print_Profile() {
    cout << name << " " << insta_ID << endl;
}

//여기의 문제 배열들을 꼭 잡아놔야
```

110 % 문제가 검색되지 않음

출력

출력 보기 선택(S): 빌드

[>Project1.vcxproj] -> C:\Users\200596\Desktop\Project1\Debug\Project1.exe

[>Project1.vcxproj] 프로젝트를 빌드했습니다.

***** 빌드: 1개 성공, 0개 실패, 0개 경고 *****

***** 빌드(73) 오후 7:46에 시작되었고 01:04:2 초(73) 소요됨 *****

오류 목록 출력

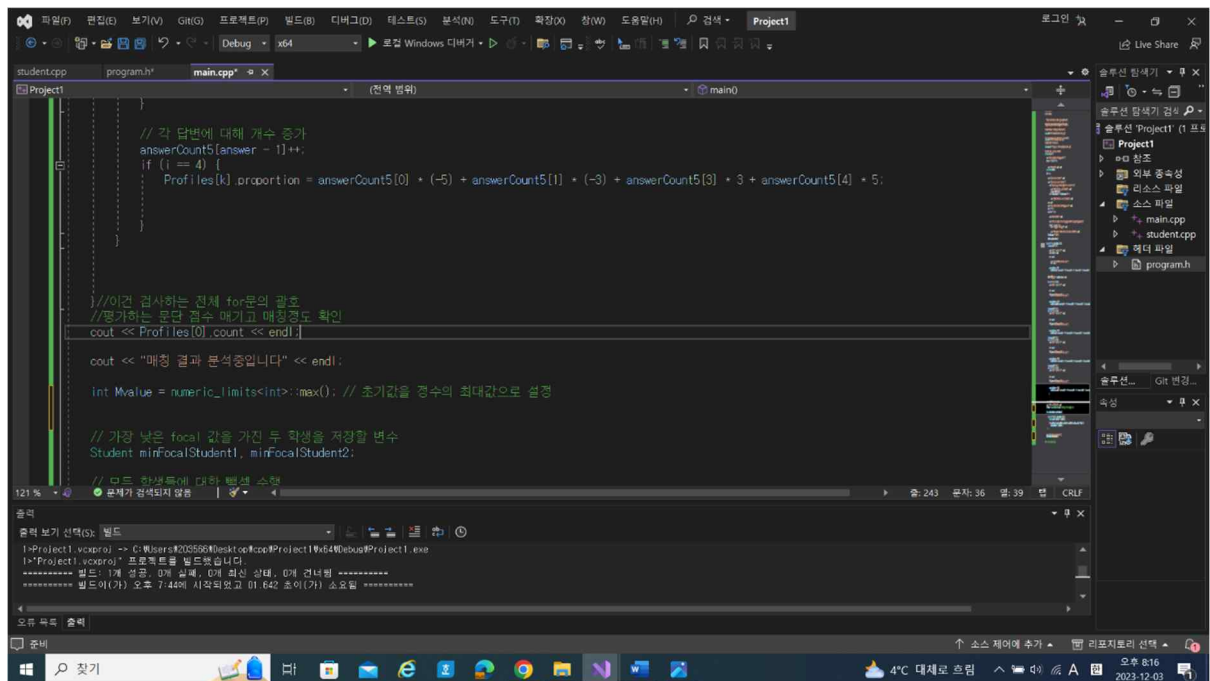
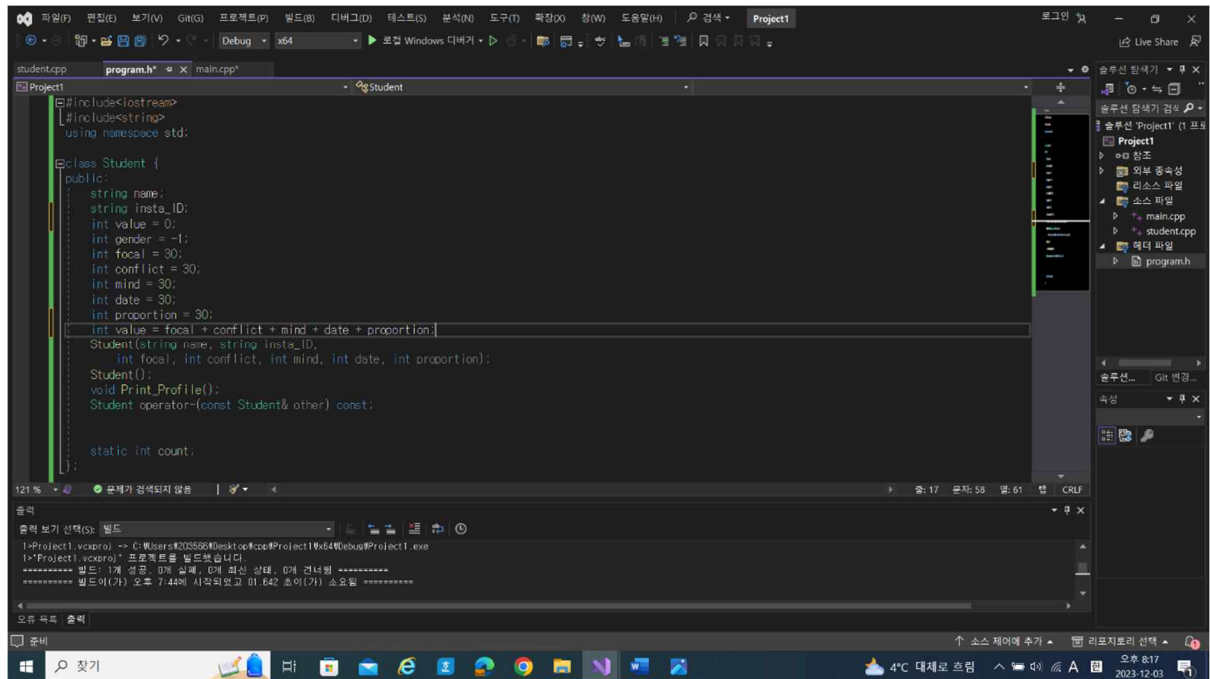
준비

찾기

4°C 대세로 으름

오후 8:17

2023-12-03

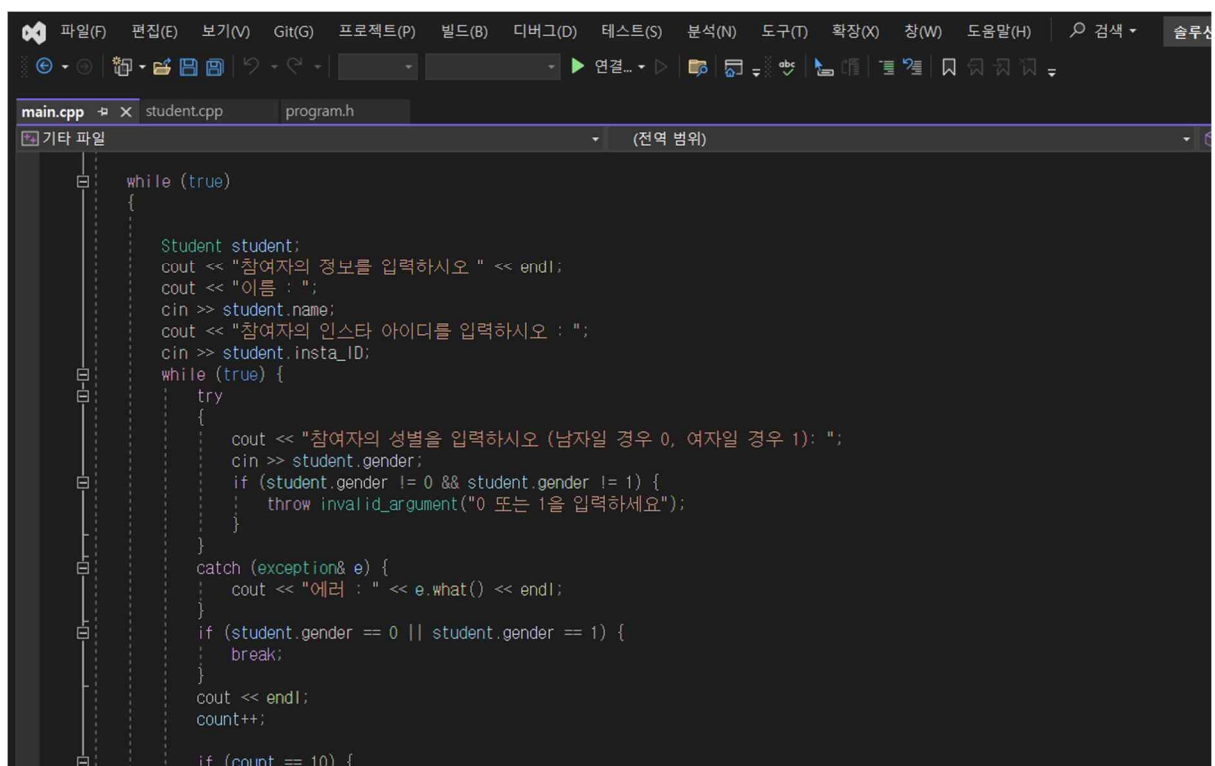


(6) 기능1 수정 (학생들의 성별을 입력 받는 코드 수정)

- 입출력 : 학생들에게 성별에 관한 정보를 입력 받음
- 설명 : 원래 학생들에게 입력 받을 때의 사용했던 반복문을 수정
- 적용된 배운 내용

1. try catch블록을 통해 잘못 입력되었을 때를 예외로 두는 예외처리 코드를 작성하였다.

-코드 스크린 샷-



```
while (true)
{
    Student student;
    cout << "참여자의 정보를 입력하십시오 " << endl;
    cout << "이름 : ";
    cin >> student.name;
    cout << "참여자의 인스타 아이디를 입력하십시오 : ";
    cin >> student.insta_ID;
    while (true) {
        try
        {
            cout << "참여자의 성별을 입력하십시오 (남자일 경우 0, 여자일 경우 1): ";
            cin >> student.gender;
            if (student.gender != 0 && student.gender != 1) {
                throw invalid_argument("0 또는 1을 입력하세요");
            }
        }
        catch (exception& e) {
            cout << "에러 : " << e.what() << endl;
        }
        if (student.gender == 0 || student.gender == 1) {
            break;
        }
        cout << endl;
        count++;
    }
    if (count == 10) {
```

(7) 추가 기능 (채점 기준 파일 생성)

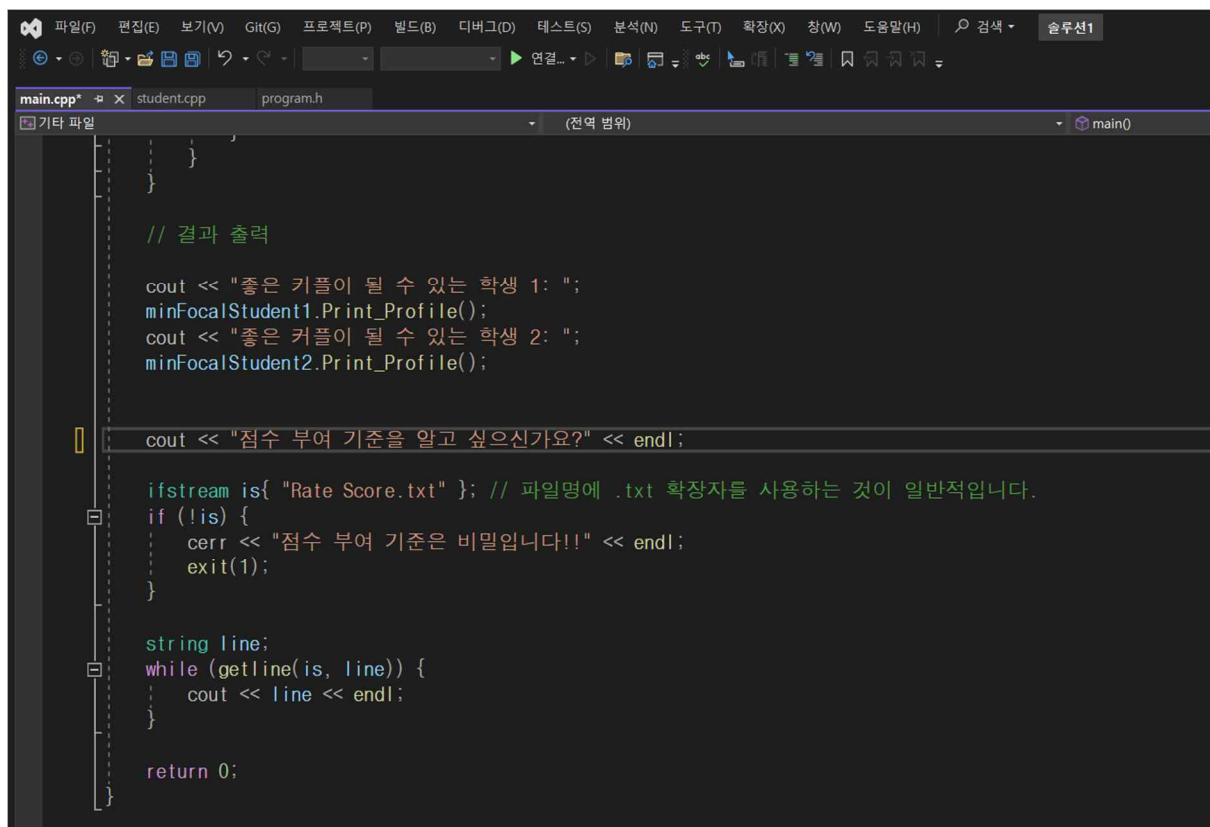
- 입출력 : 채점기준과 관련된 정보를 적어놓은 파일을 출력
- 설명 : 검사가 끝난 후 점수가 어떻게 산정되는지에 대한 관련 정보를 적어놓은 파일을 출력한다.

- 적용된 배운 내용

1. ifstream을 이용하여 Rate score.txt 파일을 읽는다.

2. cerr이라는 표준 에러 출력시스템을 사용하여 에러 메시지 출력 시 사용한다.

-코드 스크린 샷-



```
main.cpp* x student.cpp program.h
기타 파일 (전역 범위) main0

}
}

// 결과 출력

cout << "좋은 커플이 될 수 있는 학생 1: ";
minFocalStudent1.Print_Profile();
cout << "좋은 커플이 될 수 있는 학생 2: ";
minFocalStudent2.Print_Profile();

cout << "점수 부여 기준을 알고 싶으신가요?" << endl;

ifstream is{ "Rate Score.txt" }; // 파일명에 .txt 확장자를 사용하는 것이 일반적입니다.
if (!is) {
    cerr << "점수 부여 기준은 비밀입니다!!" << endl;
    exit(1);
}

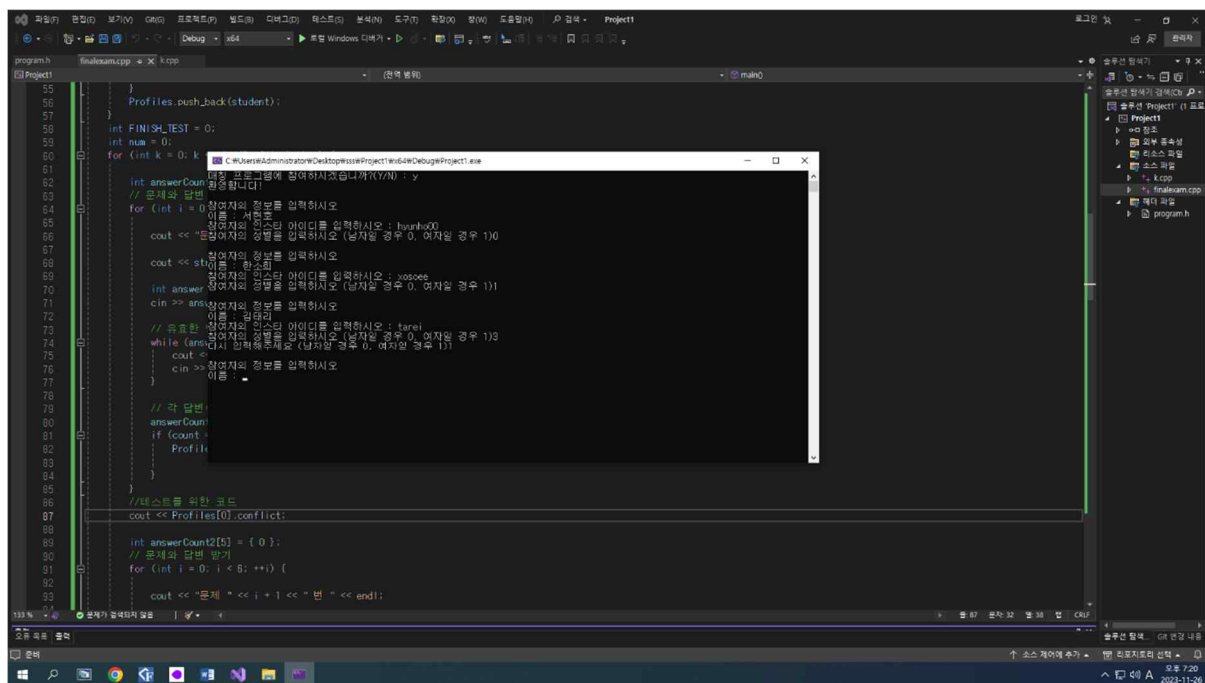
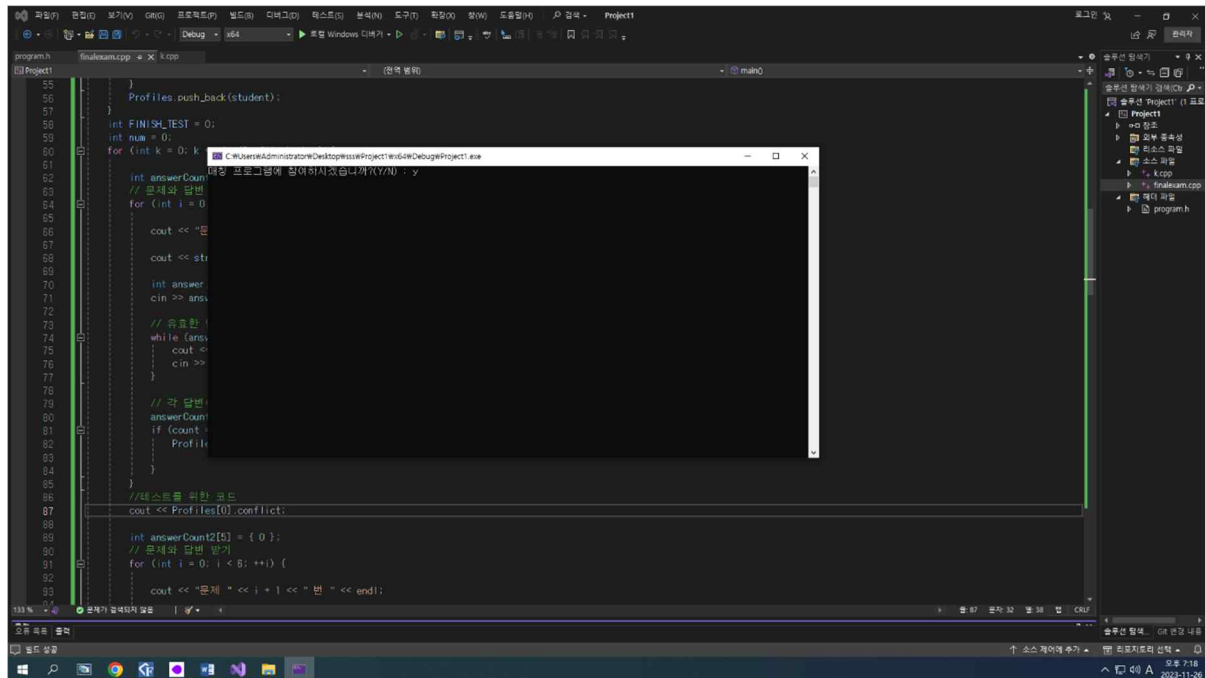
string line;
while (getline(is, line)) {
    cout << line << endl;
}

return 0;
}
```

2) 테스트 결과

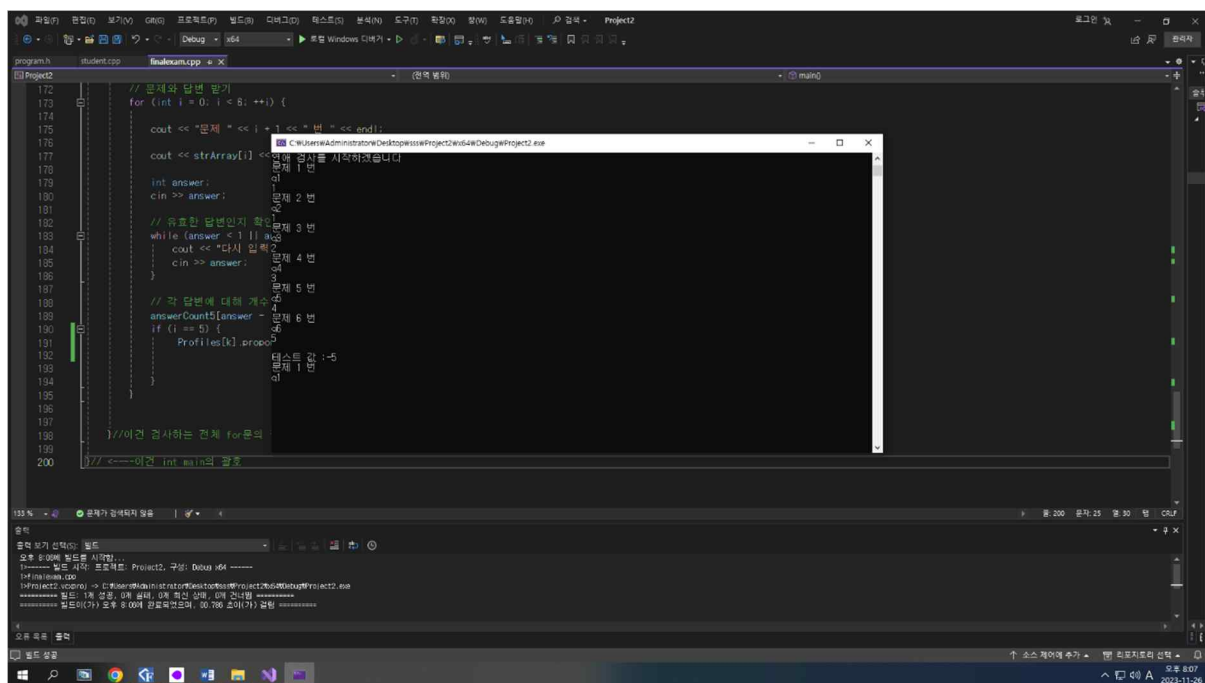
(1) 기능1(프로그램에 참여한 학생들의 정보를 입력받기)

- 설명 : 학생들의 개인정보를 입력 받고 잘못된 입력 값을 받은 경우 수정을 제안함
- 테스트 결과 스크린샷



(2) 기능2(학생들에게 입력 받은 데이터를 통해 5가지 항목에 대한 점수를 매기고 계산한다.)

- 설명 : 5가지 항목에 대한 점수를 매기고 여기서는 conflict 항목의 점수를 테스트 하는 코드를 추가하여 테스트 함 점수 (계산은 다음과 같음 : 1번 선택시 -5, 2번 선택시 -3, 3번 선택시 0, 4번 선택시 +3, 5번 선택시 +5)
- 테스트 결과 스크린샷



The screenshot shows a C++ IDE with a source file named `student.cpp` and a console window displaying the program's output. The code in `student.cpp` includes a loop for processing 5 items, a while loop for input validation, and a switch statement for calculating scores based on user input. The console output shows the program starting, asking for item numbers, and calculating the score for the 'conflict' item based on the input '1'.

```
// 문제와 답변 찾기
for (int i = 0; i < 5; ++i) {
    cout << "문제 " << i + 1 << "번 " << endl;
    cout << strArray[i] << endl;
    int answer;
    cin >> answer;

    // 유효한 답변인지 확인
    while (answer < 1 || answer > 5) {
        cout << "다시 입력하세요" << endl;
        cin >> answer;
    }

    // 각 답변에 대한 점수
    answerCounts[answer]++;

    if (i == 4) {
        Profiles[k].propose();
    }
}

//이전 int main의 끝
```

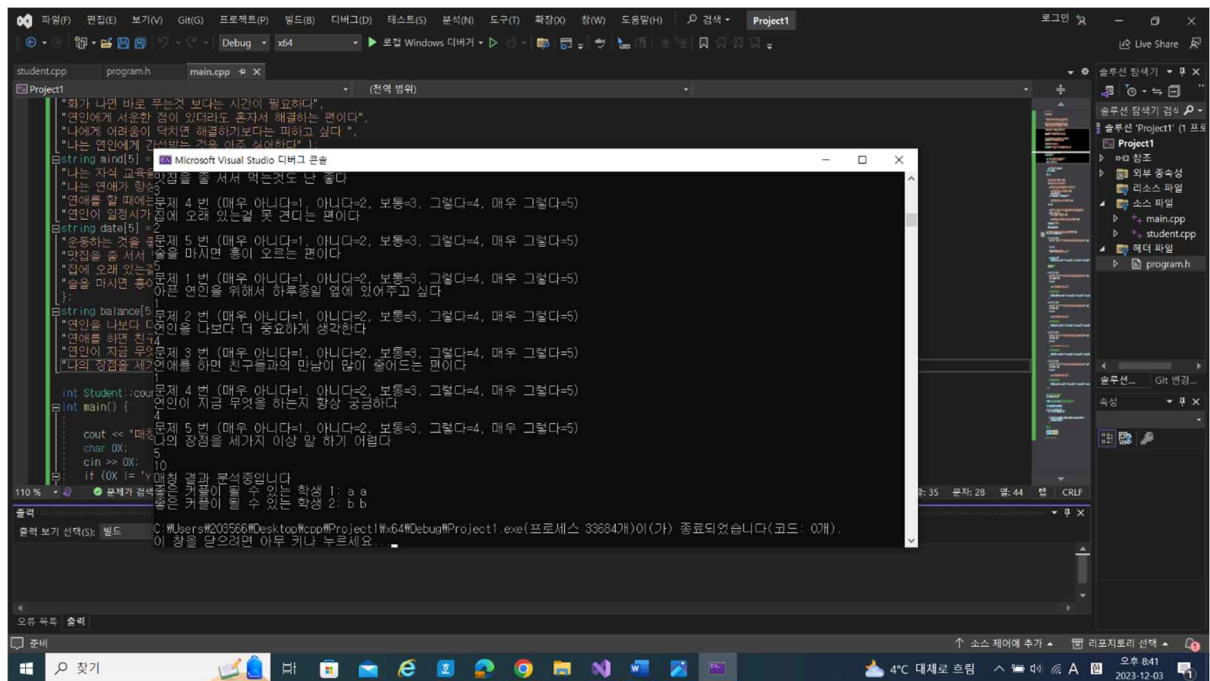
Console Output:

```
문제 1번
문제 2번
문제 3번
문제 4번
문제 5번
문제 6번
점수: 1번 -5
점수: 2번 -3
점수: 3번 0
점수: 4번 +3
점수: 5번 +5
```

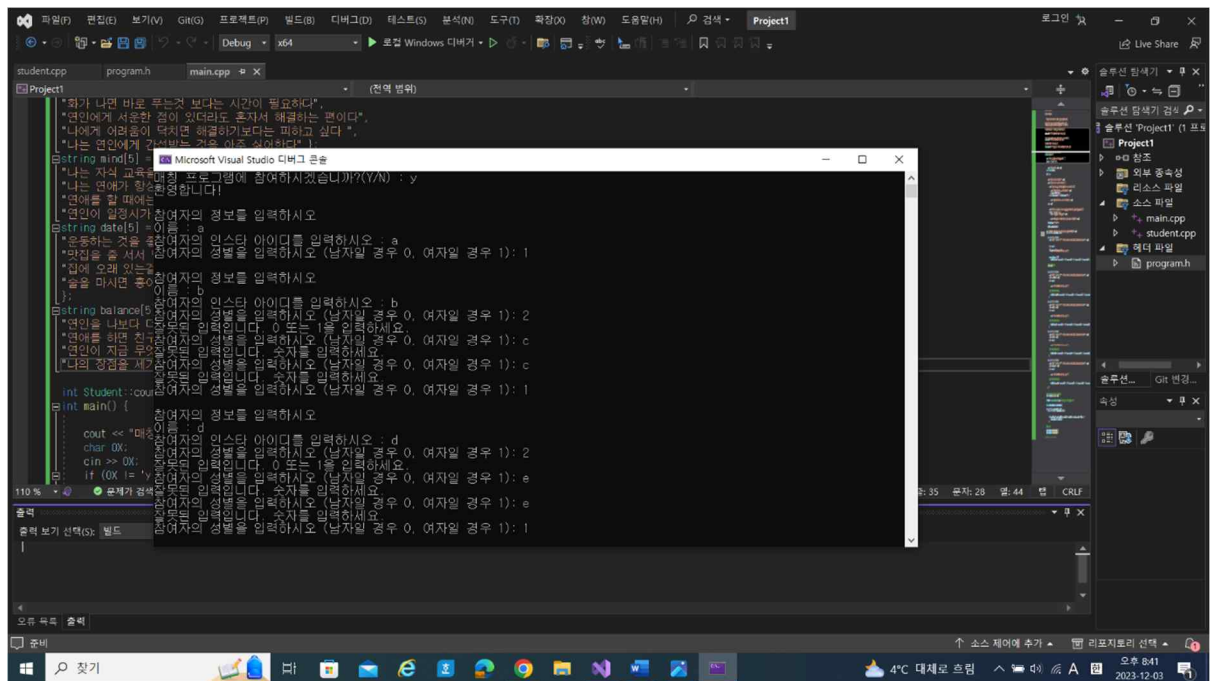

(3) 기능2(학생들에게 할 질문 선정)

- 설명 : 커플 선정에 필요한 데이터를 얻기 위해 할 질문들을 선정 후 학생들에게 질문한 후 답변을 저장한다

-테스트 결과 스크린 샷-



```
student.cpp  program.h  main.cpp  X
Project1
*화가 나면 바로 무는것 보다는 시간이 필요하다*.
*연인에게 서운한 점이 있다면도 혼자서 해결하는 편이다*.
*나에게 어려움이 닥치면 해결하기보다는 피하고 싶다 *.
*나는 연인에게 간섭하는 것을 아주 싫어한다 *.
String mind[5] = {"Microsoft Visual Studio 디버그 콘솔"}
*나는 자식 교육에 관심을 잘 서서 먹는것도 난 좋다
*나는 연애가 힘들
문제 4 번 (매우 아니다=1, 아니다=2, 보통=3, 그렇다=4, 매우 그렇다=5)
*연인이 걱정사가 많이 오래 있는일 못 견디는 편이다
String date[5] = {""}
*응답하는 것을 주 문제 5 번 (매우 아니다=1, 아니다=2, 보통=3, 그렇다=4, 매우 그렇다=5)
*관심을 잘 서서 술을 마시면 힘이 오는 편이다
*집에 오래 있는것 문제 1 번 (매우 아니다=1, 아니다=2, 보통=3, 그렇다=4, 매우 그렇다=5)
술을 마시면 좋아 연인을 위해서 하루종일 옆에 있어주고 싶다
String balance[5] = {""}
*연인을 나보다 더 연인을 나보다 더 중요하게 생각한다
*연애를 하면 친구
*연인이 지금 무언가를 3 번 (매우 아니다=1, 아니다=2, 보통=3, 그렇다=4, 매우 그렇다=5)
*나의 경점을 제2연애를 하면 친구들과의 만남이 많이 줄어드는 편이다
int Student::count 문제 4 번 (매우 아니다=1, 아니다=2, 보통=3, 그렇다=4, 매우 그렇다=5)
int main() {
4
cout << "대
char 0X;
cin >> 0X;
if (0X != 'y')
cout << "문제가 없습니다
출력
출력 보기 선택(s): 디버그 C:\Users\20356\Desktop\cpp\Project1\Win64\Debug\Project1.exe(프로세스 33684개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```



```
student.cpp  program.h  main.cpp  X
Project1
*화가 나면 바로 무는것 보다는 시간이 필요하다*.
*연인에게 서운한 점이 있다면도 혼자서 해결하는 편이다*.
*나에게 어려움이 닥치면 해결하기보다는 피하고 싶다 *.
*나는 연인에게 간섭하는 것을 아주 싫어한다 *.
String mind[5] = {"Microsoft Visual Studio 디버그 콘솔"}
*나는 자식 교육에 관심을 잘 서서 먹는것도 난 좋다
*나는 연애가 힘들
문제 4 번 (매우 아니다=1, 아니다=2, 보통=3, 그렇다=4, 매우 그렇다=5)
*연인이 걱정사가 많이 오래 있는일 못 견디는 편이다
String date[5] = {""}
*응답하는 것을 주 문제 5 번 (매우 아니다=1, 아니다=2, 보통=3, 그렇다=4, 매우 그렇다=5)
*관심을 잘 서서 술을 마시면 힘이 오는 편이다
*집에 오래 있는것 문제 1 번 (매우 아니다=1, 아니다=2, 보통=3, 그렇다=4, 매우 그렇다=5)
술을 마시면 좋아 연인을 위해서 하루종일 옆에 있어주고 싶다
String balance[5] = {""}
*연인을 나보다 더 연인을 나보다 더 중요하게 생각한다
*연애를 하면 친구
*연인이 지금 무언가를 3 번 (매우 아니다=1, 아니다=2, 보통=3, 그렇다=4, 매우 그렇다=5)
*나의 경점을 제2연애를 하면 친구들과의 만남이 많이 줄어드는 편이다
int Student::count 문제 4 번 (매우 아니다=1, 아니다=2, 보통=3, 그렇다=4, 매우 그렇다=5)
int main() {
4
cout << "대
char 0X;
cin >> 0X;
if (0X != 'y')
cout << "문제가 없습니다
출력
출력 보기 선택(s): 디버그 C:\Users\20356\Desktop\cpp\Project1\Win64\Debug\Project1.exe(프로세스 33684개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```


(4) 기능3 (학생들에게 받은 데이터를 통해 점수를 계산)

- 입력 : 학생들의 답변 결과로 얻은 점수, 점수들끼리의 결합
 - 설명 : 학생들의 답변 결과로 생성된 5가지의 항목에 대한 점수들을 특정 방법으로 계산하여 좋은 커플이 될 수 있는 가능성이 가장 높은 두 명을 선정하여 뽑았다.
- 테스트 결과 스크린 샷-

```
Microsoft Visual Studio 디버그 콘솔
문제 4 번 (매우 아니다=1, 아니다=2, 보통=3, 그렇다=4, 매우 그렇다=5)
집에 오래 있는걸 못 견디는 편이다
1
문제 5 번 (매우 아니다=1, 아니다=2, 보통=3, 그렇다=4, 매우 그렇다=5)
술을 마시면 흥이 오르는 편이다
2
문제 1 번 (매우 아니다=1, 아니다=2, 보통=3, 그렇다=4, 매우 그렇다=5)
아픈 연인을 위해서 하루종일 옆에 있어주고 싶다
2
문제 2 번 (매우 아니다=1, 아니다=2, 보통=3, 그렇다=4, 매우 그렇다=5)
연인을 나보다 더 중요하게 생각한다
2
문제 3 번 (매우 아니다=1, 아니다=2, 보통=3, 그렇다=4, 매우 그렇다=5)
연애를 하면 친구들과의 만남이 많이 줄어드는 편이다
1
문제 4 번 (매우 아니다=1, 아니다=2, 보통=3, 그렇다=4, 매우 그렇다=5)
연인이 지금 무엇을 하는지 항상 궁금하다
2
문제 5 번 (매우 아니다=1, 아니다=2, 보통=3, 그렇다=4, 매우 그렇다=5)
나의 장점을 세가지 이상 말 하기 어렵다
2
10
매칭 결과 분석중입니다
(좋은 커플이 될 수 있는 학생 1: seo hseksl12
(좋은 커플이 될 수 있는 학생 2: kim sdjfs1
점수 부여 기준을 알려드리겠습니다
?
```

(5) 추가 기능 (채점 기준 파일 생성)

- 설명 : 검사가 끝난 후 점수 산정기준 관련 정보를 적어놓은 파일을 출력한다.

-테스트 결과 스크린 샷-

```
점수 부여 기준을 알려드리겠습니다
Do you want to know more?
if you want to know details, send email (adress : hhseo9519@naver.com)

C:\Users\203566\Desktop\CPP 기말고사 대체과제\Project12\x64\Debug\Project12.exe
코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

4. 계획 대비 변경 사항

1) 일정 순서 변경 #1

- 이전

: 학생들에게 줄 문제를 선정하는 것을 11/24까지 완료하기로 함.

12/8 까지 점수 채점을 완료하기로 함

- 이후 : 점수 채점 코드를 먼저 짰 후 문제 선별을 12/8일 까지하기로 함

- 사유 : 촉박한 기말시험 일정으로 시간이 더 많이 드는 것을 먼저 수행하기로 함.

2) 일정 순서 변경 #2

- 이전

: 처음에 기획한 기능들을 수행하기로 함

- 이후

:여러가지 추가 기능들을 만들어서 추가함.

- 사유 : 프로젝트의 완성도 수준 함양

5. 프로젝트 일정

(진행한 작업과 진행 중인 작업 등을 표기)

업무		11/3	11/10	11/17	11/24
제안서 작성		완료			
기능1	세부기능1		완료		
	세부기능2			완료	
기능2					완료
기능3					
기능4	세부기능1				
기능5	세부기능2			----->	
날짜	12/1	12/8	12/15	12/22
기능1	세부기능1				
	세부기능2				
기능2		완료			
기능3		완료			
기능4		----->			
기능5		----->			